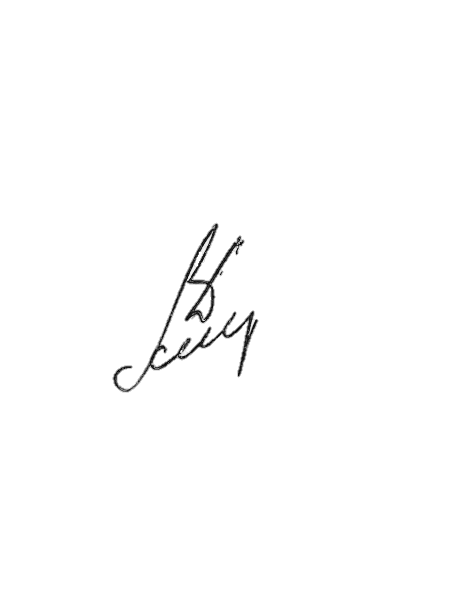
**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Национальный исследовательский ядерный университет «МИФИ»**

**ОТЧЕТ**

по экзаменационному проекту

по дисциплине: «Теория и технология программирования»



Выполнил: *студент группы Б20-901*  Кондрашкина К.Д.

Проверил: Смирнов Д.С.

Москва 2023 г

**Оглавление**

[**1. Описание задания:** 3](#_Toc138287746)

[**2. ER-диаграмма основных сущностей предметной области:** 5](#_Toc138287747)

[**3. Use-case диаграмма:** 6](#_Toc138287748)

[**4. Диаграмма классов UML:** 7](#_Toc138287749)

[**5. Wireframe-эскизы GUI:** 17](#_Toc138287750)

[**6. Скриншоты реализованного интерфейса:** 23](#_Toc138287751)

[**7. Описание функционирования реализованной программы:** 29](#_Toc138287752)

[*Приложения А. Логи системы контроля версий:* 31](#_Toc138287753)

[*Приложение Б. Исходный код программы:* 32](#_Toc138287754)

**1. Описание задания:**

**Задание**: разработать модель аукциона в соответствии с приведенной блок схемой.

**Описание предметной области**: Система имитирует модель английского аукциона, которая основывается на эмоциях и в принципах на человеческой психологии.

Для каждого игрока описаны постоянные факторы, и ранжированы от 1 до 10. (см. Приложение лист «Качества игроков»)

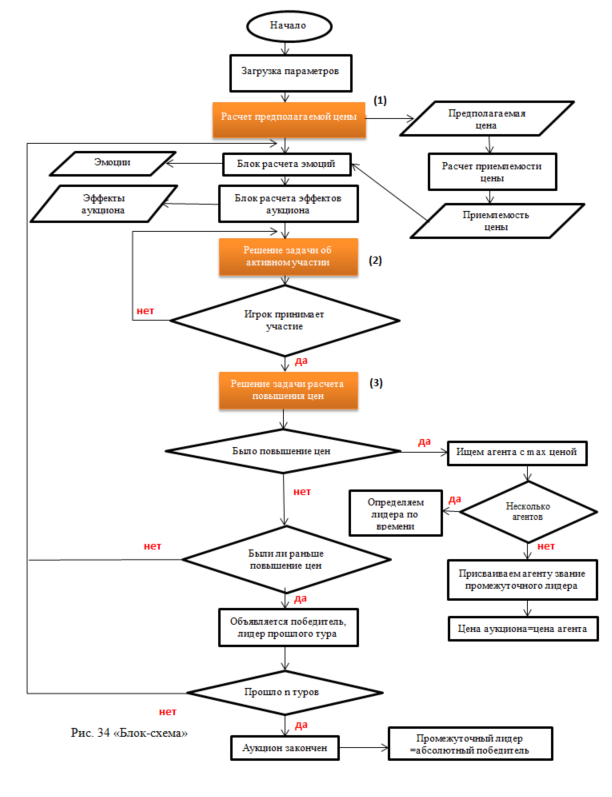
Аукцион состоит из 50 лотов – монет. Каждая монета обладает своими качествами. Все характеристики товара ранжированы от 1 до 10. (см. Приложение лист «Качества товаров»)

Влияющие факторы на поведение участника аукциона:

1. Качества товара
   1. Редкость товара
   2. Известность товара
   3. Значимость товара на рынке
2. Качества игроков
   1. Типа игрока
      1. Спекулянт — лицо, покупающее что-либо с целью перепродажи и получения выгоды, а не для собственного потребления
      2. Коллекционер – лицо, покупающее товар для пополнения своей коллекции
   2. Благосостояние
   3. Скупость
   4. Склонность к риску
   5. Самоуверенность
   6. Необходимость в товаре
3. Эмоции
   1. Страх бедности – страх, который испытывает человек при мысли о том, что после покупки товара, он обеднеет.
   2. Страх потери – страх того, что игрок останется без желаемого товара, что он упустит возможность приобретения.
   3. Азарт
   4. Неуверенность в товаре – это все сомнения в товаре, которые испытывает человек, во время торгов.
   5. Уверенность в себе – уверенность в своих возможностях выиграть торги
4. Эффекты аукциона
   1. Количество игроков, участвующих в аукционе
   2. Активность игроков
   3. Разность текущей цены и стартовой цены
   4. Рост аукциона – насколько быстро растет цена во время аукциона

|  |  |
| --- | --- |
| **Постоянные факторы** | **Изменчивые факторы** |
| Редкость товара | Разность цен |
| Известность товара | Рост аукциона |
| Значимость товара на рынке | **Изменчивые параметры** |
| Деньги | Приемлемость цены |
| Необходимость в товаре | Страх бедности |
| Скупость | Неуверенность в товаре |
| Тип игрока | Уверенность в себе |
| Самоуверенность | Азарт |
| Склонность к риску | Страх потери |
| Активность | Предполагаемая цена |

Таблица «Факторы»



Блок-схема

Математическое описание см. Приложение 2

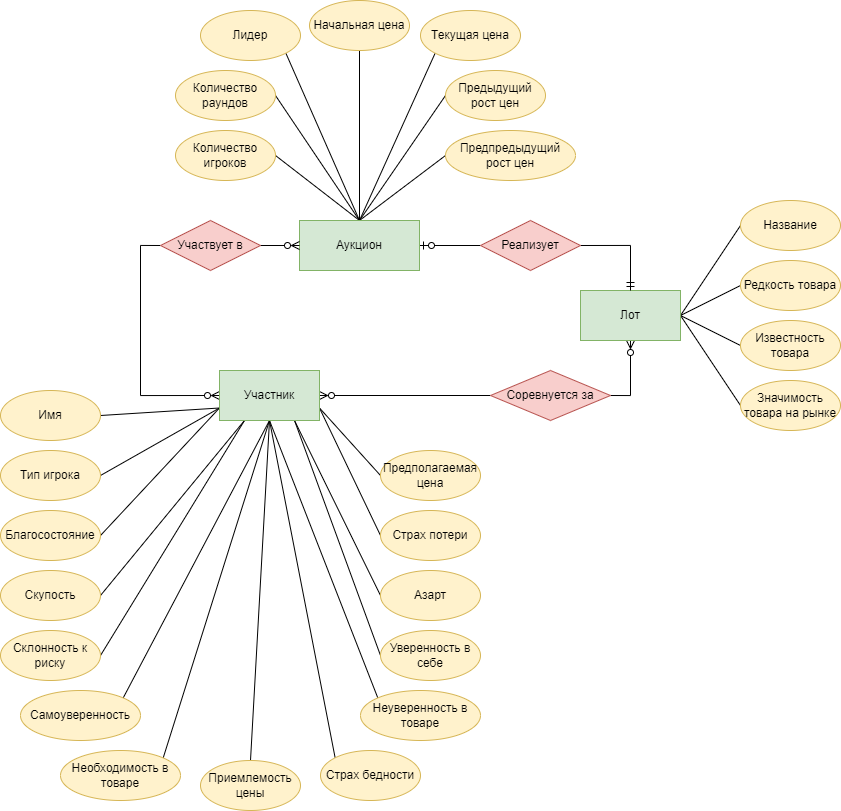
**Требования к пользовательскому интерфейсу**

* Пользователь должен иметь возможность просмотреть параметры игрока
* Пользователю должен выводится график цены (по оси Х – раунды аукциона)

**Дополнительный функционал**

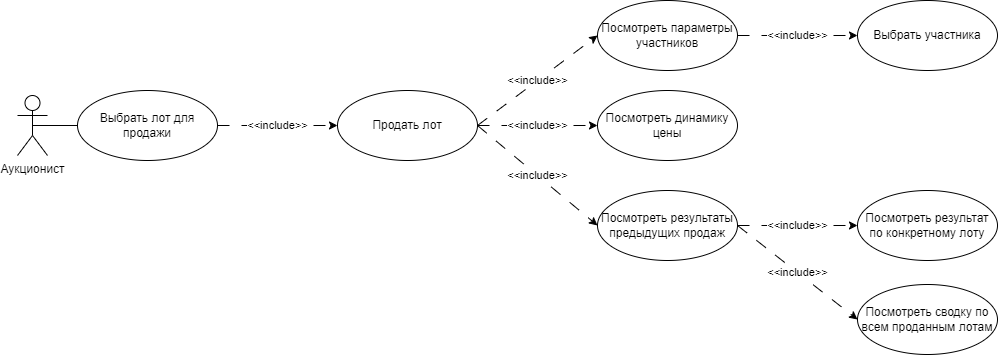
* Реализовать проведение серии торгов

**2. ER-диаграмма основных сущностей предметной области:**

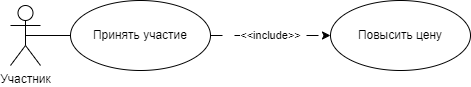


Участник аукциона может участвовать в 0 или нескольких аукционах (аукцион = продажа одного лота), в свою очередь в одном аукционе может принимать участие 0 или несколько участников. Конкретный участник может соревноваться за несколько лотов или не за один, также как и за конкретный лот может соревноваться 0 или несколько участников аукциона. У лота может быть 0 или 1 аукцион (его либо выставили на продажу, либо нет), в одном аукционе может быть продан один и только один лот.

**3. Use-case диаграмма:**



Аукционист [пользователь] может выбрать лот для продажи, это включает в себя процедуру продажи лота. В моменте продажу доступны такие действия, как посмотреть параметры участников аукциона, посмотреть динамику цены и посмотреть результаты предыдущих продаж. При просмотре параметров участника необходимо выбрать конкретного участника. При просмотре результатов предыдущих продаж есть возможность выбора между просмотром результатов продажи конкретного лота и таблицы-сводки по всем проданным лотам.



У участника аукциона (сущности внутри программы) в качестве действия есть только «принять участие», данное действие включает в себя поднятие цены на лот.

**4. Диаграмма классов UML:**

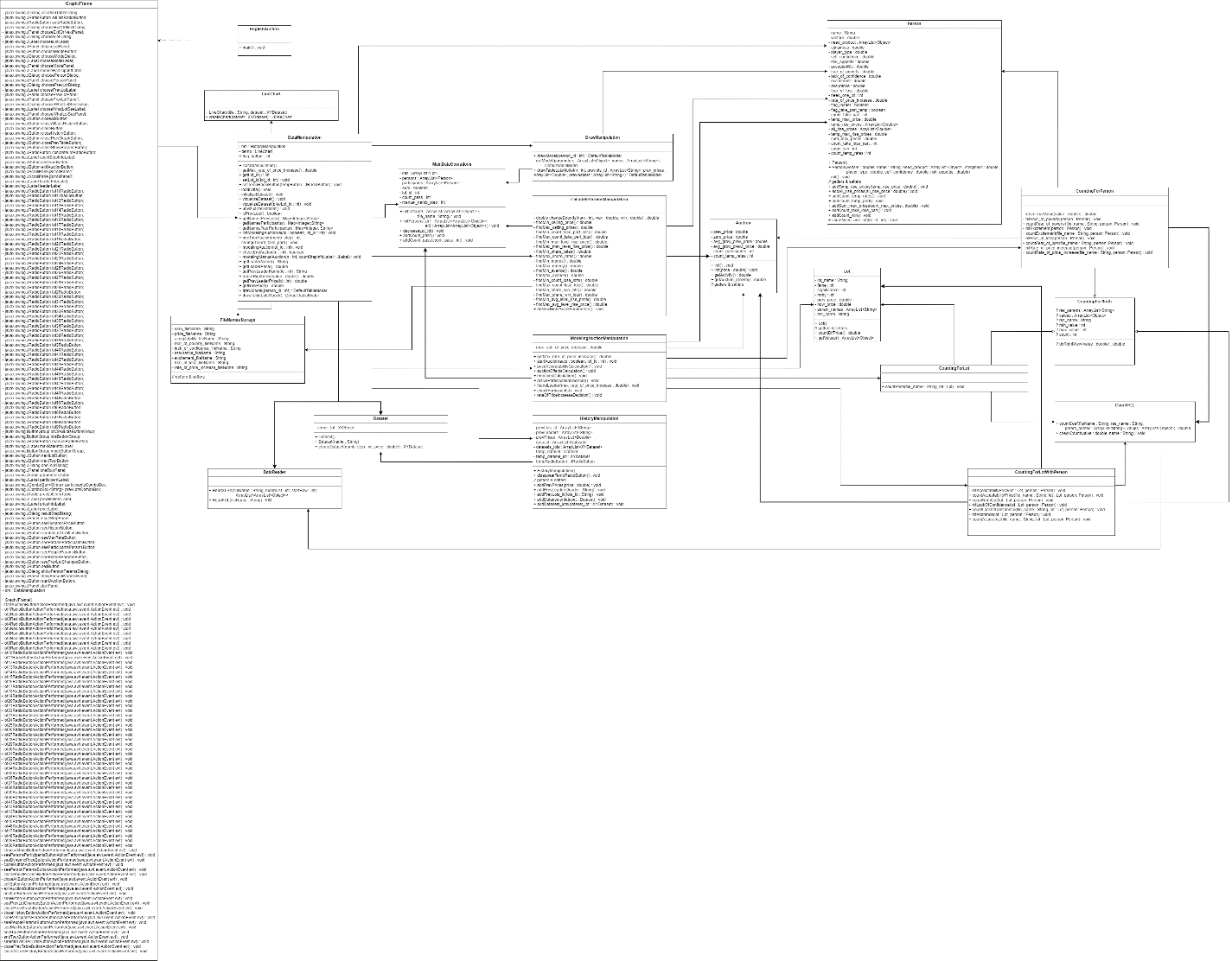
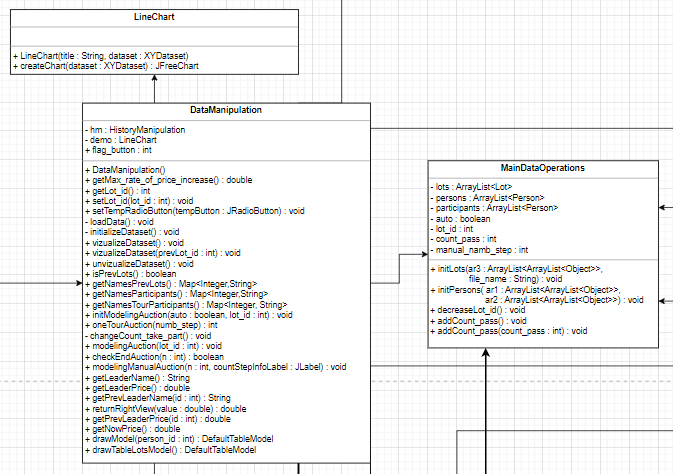


Диаграмма UML всего проекта



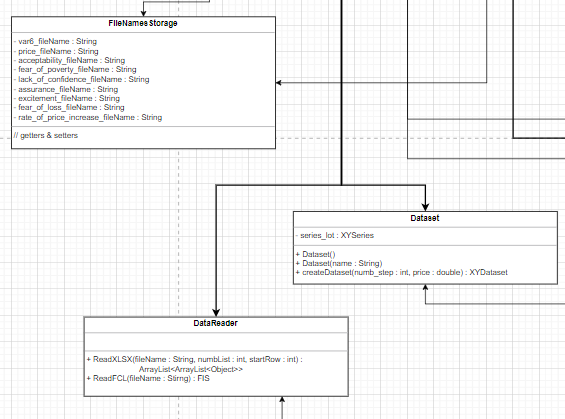
Класс GraphJFrame

В данном классе содержатся графические компоненты и методы работы с ними (при нажатии или выборе). В данном классе есть связь с классом DataManipulation.



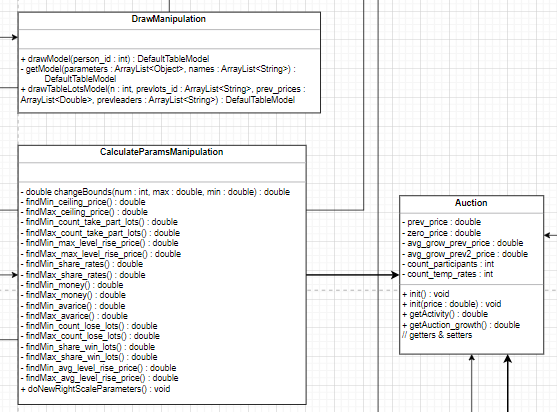
Классы DataManipulation, MainDataOperations, LineChart

В классе DataManipulation осуществляется связь действий различных классов. Этот класс отвечает за логику, которая вызывается в классе GraphJFrame. Есть методы, которые позволяют получить или учтановить величины из других классов (max\_rate\_of\_price\_increase, lot\_id и др.). Есть методы отвечающие за визуализацию графического представления динамики цены и методы, отвечающие за визуализацию в виде таблицы. В классе MainDataOperations хранятся списки с участниками аукциона, участниками текущего тура, лотами. Также в данном классе хранится флаг, который позволяет отследить какой режим выбран пользователем для проведения аукциона. Также величина, отслеживающая количество пропусков, и методы добавления количества пропусков. Присутствуют методы заполнения списков с участниками и лотами. В классе LineChart метод для графического отображения динамики цены, возвращает объект типа JFreeChart.



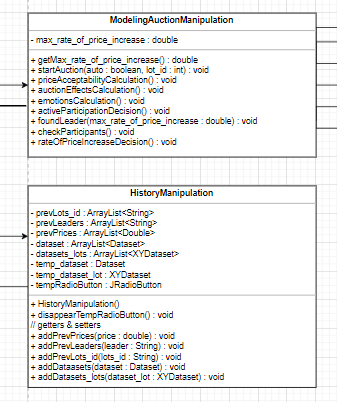
Классы FileNamesStorage, DataReader, Dataset

В классе FileNamesStorage хранятся все названия файлов, которые понадобятся в проекте. Методы получения и установления названий файлов. В классе Dataset хранится объект типа XYSeries, данный класс необходим для создания в дальнейшем объекта типа LineChart. В классе DataReader находятся методы для чтения информации из файлов (чтение из файлов форматов “xlsx” и “fcl”).



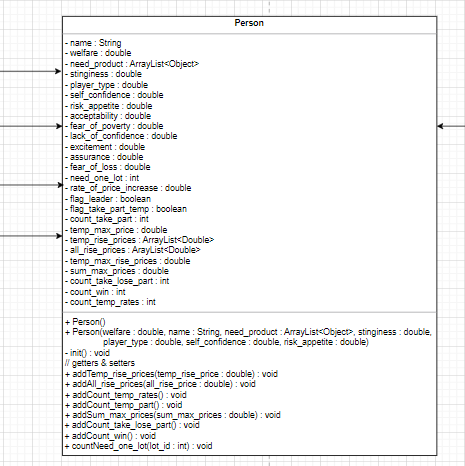
Классы CalculateParamsManipulation, DrawManipulation, Auction

В классе DrawManipulation находятся методы по заполнению таблицы и возвращению объекта типа DefaultTableModel. Методы данного класса вызываются внутри одноимённых методов класса DataManipulation. В классе Auction хранится информация о текущей продаже (начальная цена, текущая цена, предыдущий и предпредыдущий рост цен, число участников аукциона). В данном методе есть методы для расчёта эффектов аукциона (активности, роста аукциона, получение количества участников). В классе CalculateParamsManipulation находятся методы для расчёта параметров участника (тип игрока, благосостояние, скупость, желание рисковать, самоуверенность). Приватные методы данного класса позволяют найти максимальное или минимальное значение определённой величины среди всех игроков, а затем ранжировать игроков и таким образом получиться искомые параметры в границах от 0 до 10.



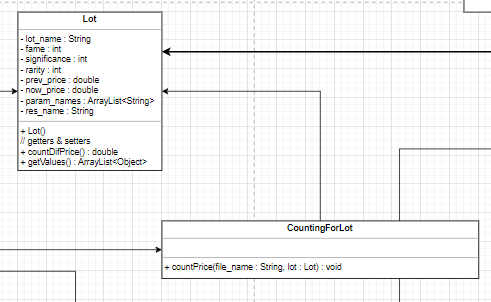
Классы ModelingAuctionManipulation, HistoryManipulation

В классе ModelingAuctionManipulation находятся методы для реализации основных этапов аукциона (старт аукциона – его инициализация, расчёт приемлемости цены, эффектов аукциона, эмоций участников аукциона, затем метод для принятия решения об участии, проверка есть ли желающие участвовать в данном раунде и метод для определения величины, на которую будет выполнен подъем цены). В классе HistoryManipulation находится информация о пройденных продажах (списки с именами проданных лотов, лидеров продаж, цен, за которые были приобретены лоты, объекты типа XYDataset и Dataset для возможности в дальнейшем визуализировать результат продажи, кнопка, которая отвечает за продаваемый в данный момент лот). В данном классе содержатся методы для добавления значений в созданные массивы и метод, отвечающий за скрытие кнопки.



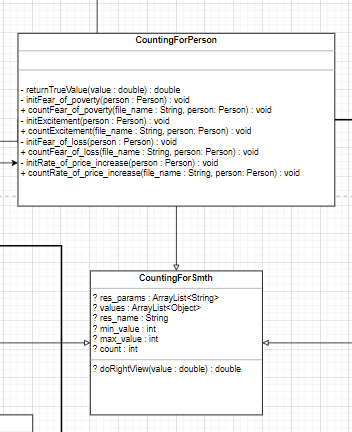
Класс Person

В данном классе хранятся параметры участника и значения его эмоций, также некоторые величины, которые необходимы для расчёта параметров участника. В классе содержатся методы для добавления величин в списки, которые содержат информацию о, например, всех поднятиях цен участником.



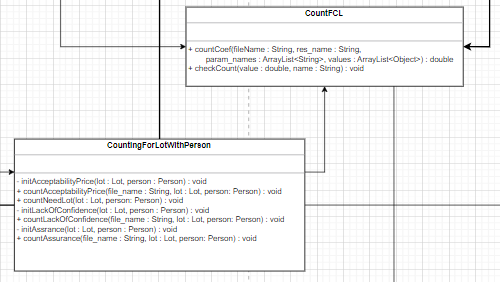
Классы Lot, CountingForLot

В классе Lot содержатся параметры конкретного лота, методы для того, чтобы установить или получить данные параметры. В классе CountingForLot находится метод, который рассчитывает начальную цену для лота.



Классы CountingForPerson, CountingForSmth

В классе CountingForSmth содержатся параметры – res\_params, которые используются для того, чтобы рассчитать итоговые величины – res\_names. Значения параметров хранятся в списке values. Для того, чтобы привести полученную величину в необходимые границы используются метод doRightView (в нём необходимы границы - min\_value и max\_value). Классы CountingForLot, CountingForPerson, CountingForLotWithPerson унаследованы от класса CountingForSmth. Класс CountingForPerson отвечает за расчёт параметров, которые не зависят от параметров лота, только от внутренних параметров человека-участника аукциона.



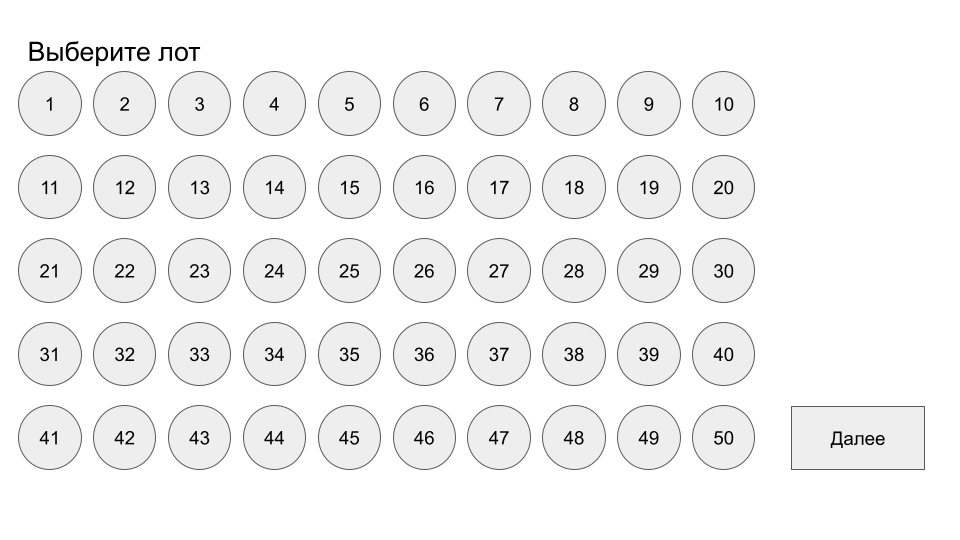
Классы CountingForLotWithPerson, CountingFCL

Класс CountingForLotWithPerson отвечает за расчёт параметров, которые зависят от параметров лота, параметров участника аукциона и эффектов аукциона. Классы CountingForLot, CountingForPerson, CountingForLotWithPerson в своих методах расчёта искомых параметров обращаются к классу CountinFCL. В классе осуществляется обращение к библиотеке jFuzzyLogic, выполняется расчёт параметра с помощью нечёткой логики и метод для проверки получено ли значение или правила для переданных параметров не были найдены.

**5. Wireframe-эскизы GUI:**

JFrame [startPanel]

Стартовая панель, которая появляется при запуске программы.

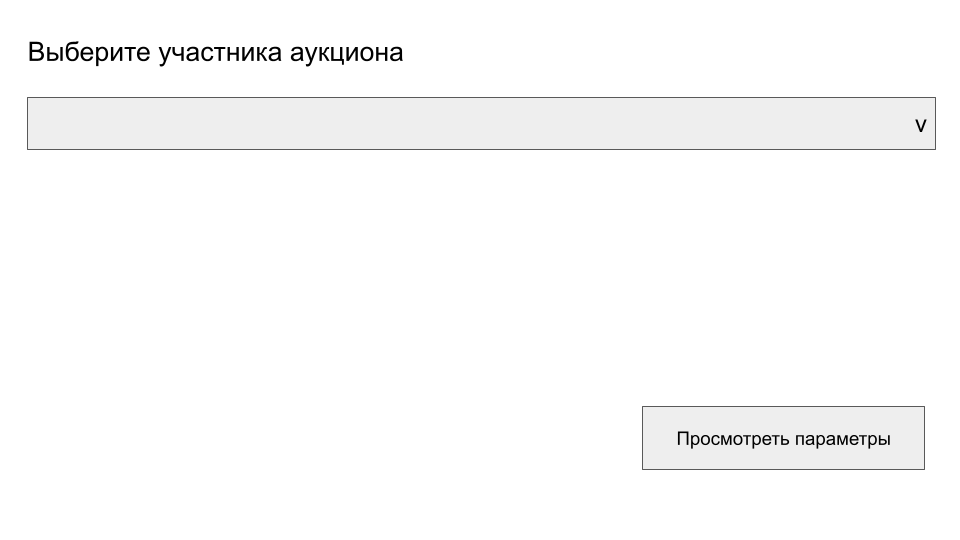
chooseLotDialog

Диалоговое окно для выбора аукционистом [пользователем] лота на продажу.



resultStepDialog

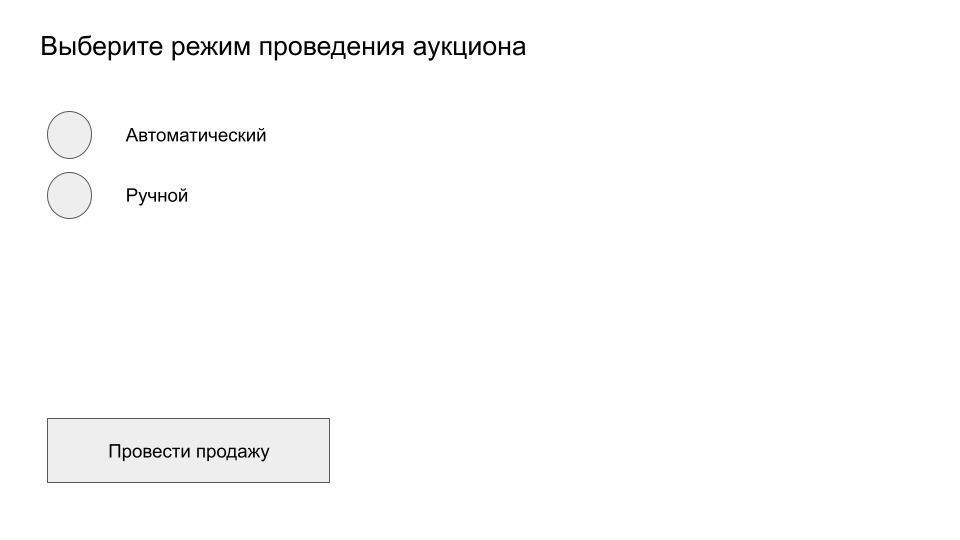
Диалоговое окно для выбора аукционистом [пользователем] действия после продажи лота. Кнопка «Закрыть результаты» становится видимой при нажатии одной из трёх первых кнопок (при просмотре параметров, динамики цены или информации о проданных лотах).

choosePersonDialog

Диалоговое окно для выбора аукционистом [пользователем] участника для просмотра его параметров (с помощью ComboBox).

showPersonParamsDialog

Диалоговое окно для того, чтобы показать параметры выбранного ранее (в диалоговом окне choosePersonDialog) участника аукциона аукционисту [пользователю].

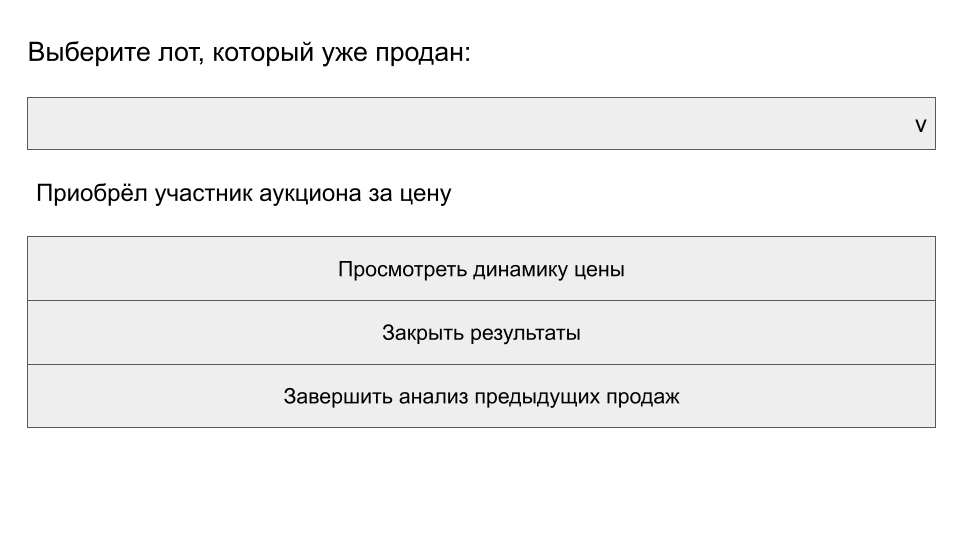


chooseModeDialog

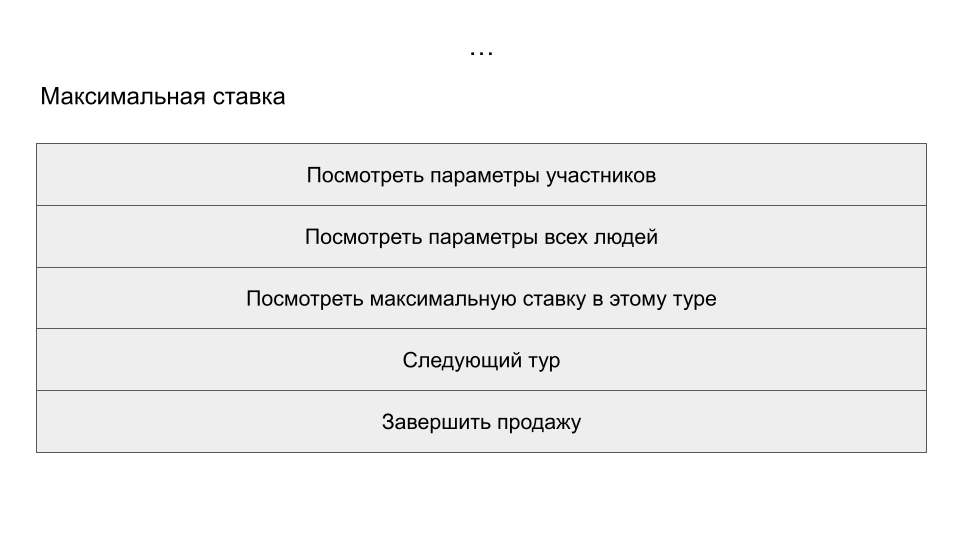
Диалоговое окно для того, чтобы аукционист [пользователь] мог выбрать режим проведения продажи лота (автоматический – программа полностью моделирует проведение продажи, ручной – аукционист моделирует продажу пошагово, отслеживая каждое поднятие цены).

chooseExitOrNextDialog

Диалоговое окно для того, чтобы аукционист [пользователь] мог выбрать продолжить аукцион (провести следующую продажу) или завершить аукцион.

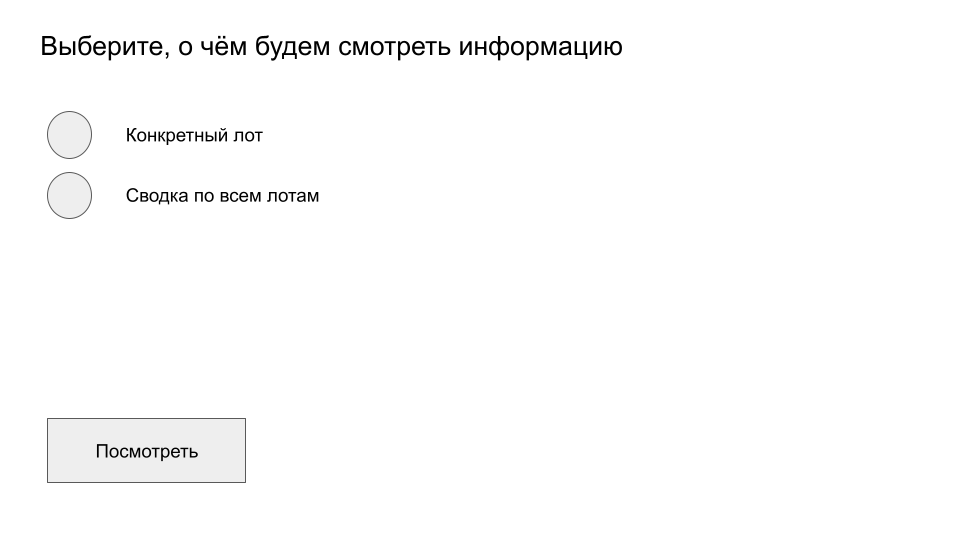
choosePrevLotDialog

Диалоговое окно для выбора аукционистом [пользователем] проданного лота для просмотра параметров его продажи (с помощью ComboBox). Кнопка «Закрыть результаты» становится видимой при нажатии кнопки «Просмотреть динамику цены».



oneTourDialog

Диалоговое окно для выбора аукционистом [пользователем] действия при ручном режиме проведения продажи (с возможностью: посмотреть параметры участников текущего тура, посмотреть параметры всех людей, заявленных в аукционе, посмотреть какая была сделана максимальная ставка в текущем туре). Кнопка «Следующий тур» видима, до тех пор, пока не пройдет несколько «пустых» туров, без поднятия цены. Кнопка «Завершить продажу» становится видимой, когда пройдёт несколько «пустых» туров. Верхние три точки («…») изменяются на текст «И раз», «И два», «Продано» в зависимости от количества туров, без поднятия цены.

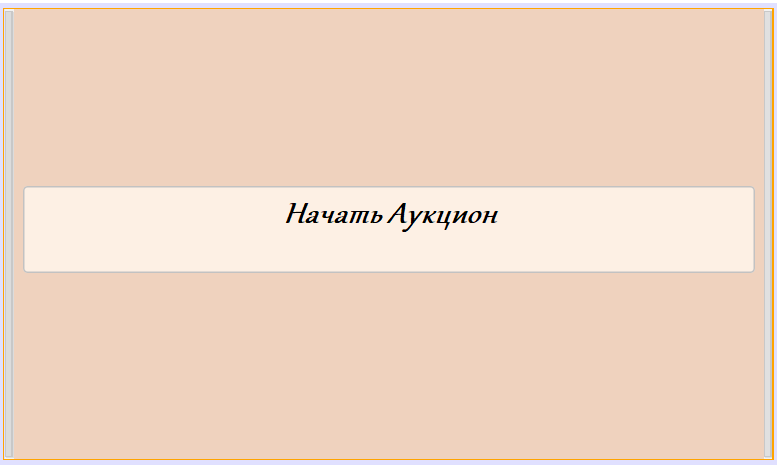
chooseWhatLotSeeDialog

Диалоговое окно для выбора аукционистом [пользователем] лотов, по которым будет отображена информация (после нажатия кнопки «Посмотреть информацию о проданных лотах» в диалоговом окне resultStepDialog).

allLotsTableDialog

Диалоговое окно для отображения аукционисту [пользователю] информации о всех проданных до этого лотах (в виде таблицы- сводки, показывающей название проданного лота, цену продажи и того, кто приобрёл данный лот).

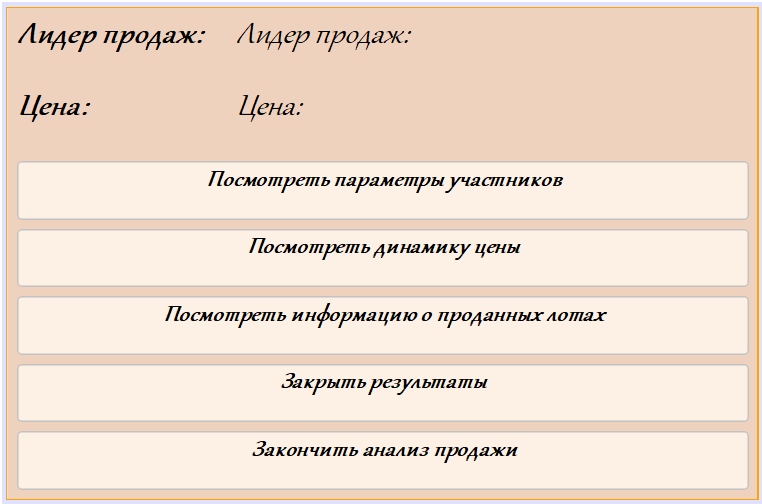
**6. Скриншоты реализованного интерфейса:**



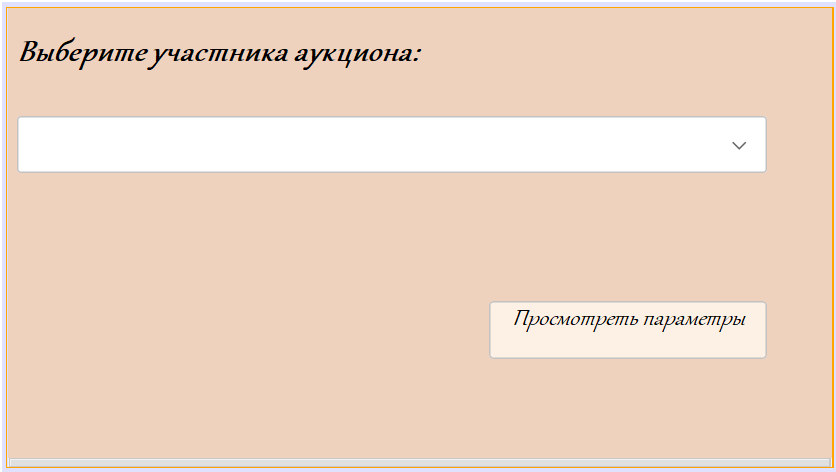
JFrame [startPanel]



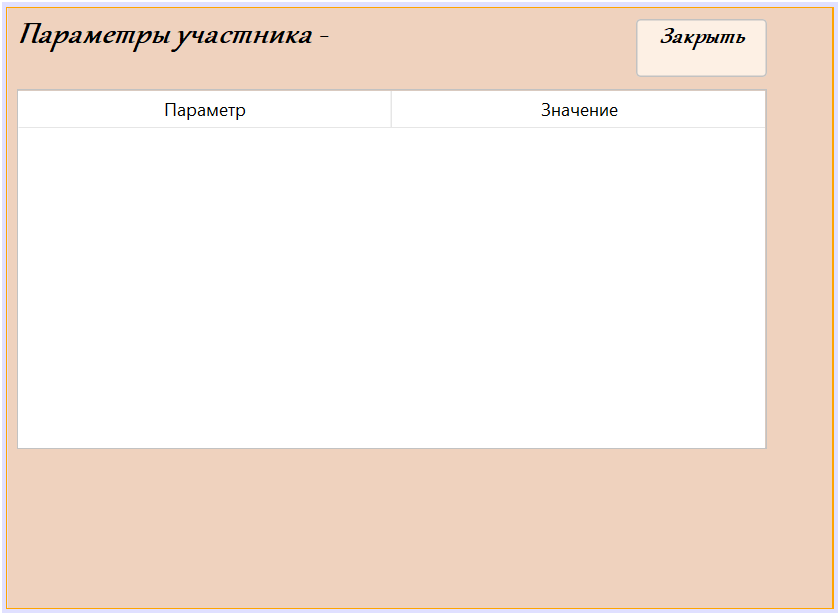
chooseLotDialog



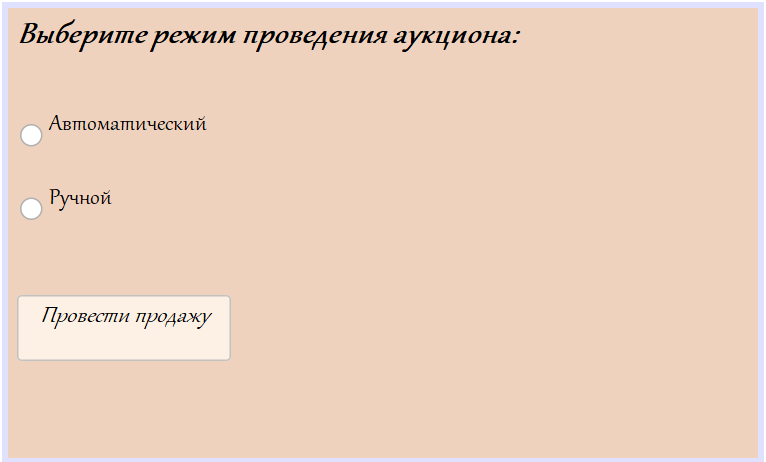
resultStepDialog



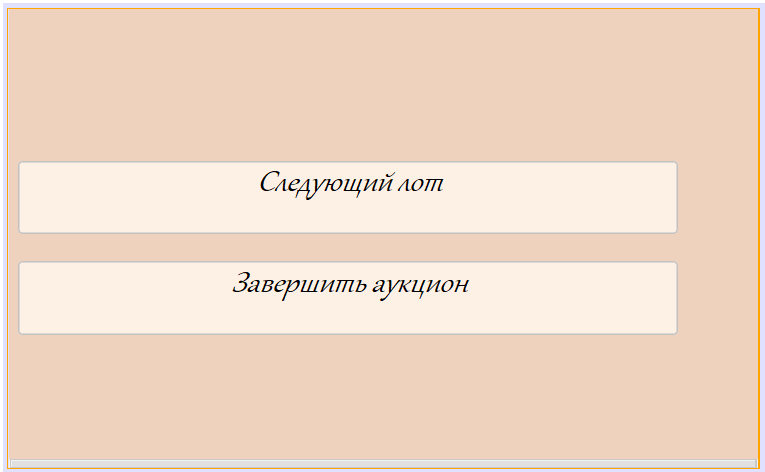
choosePersonDialog



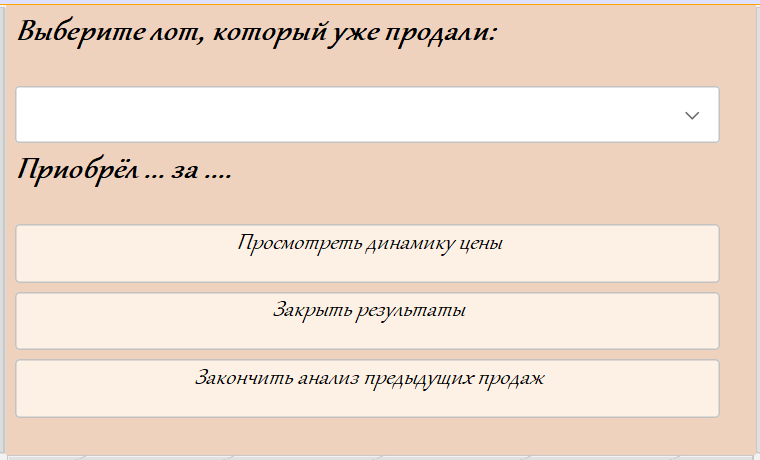
showPersonParamsDialog



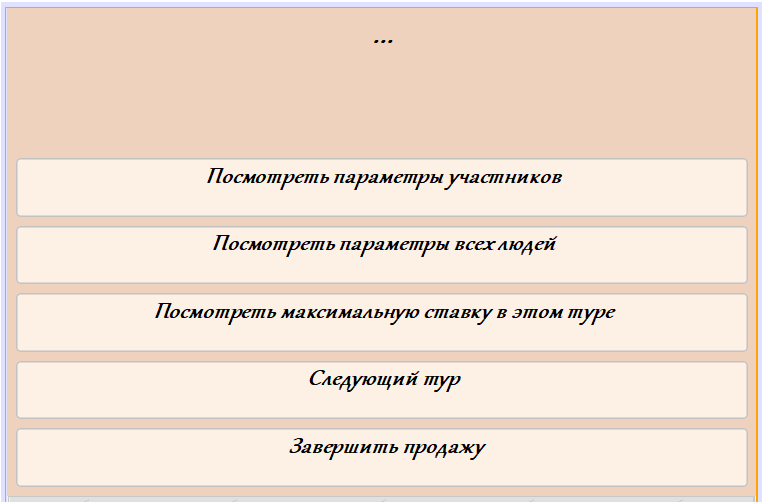
chooseModeDialog



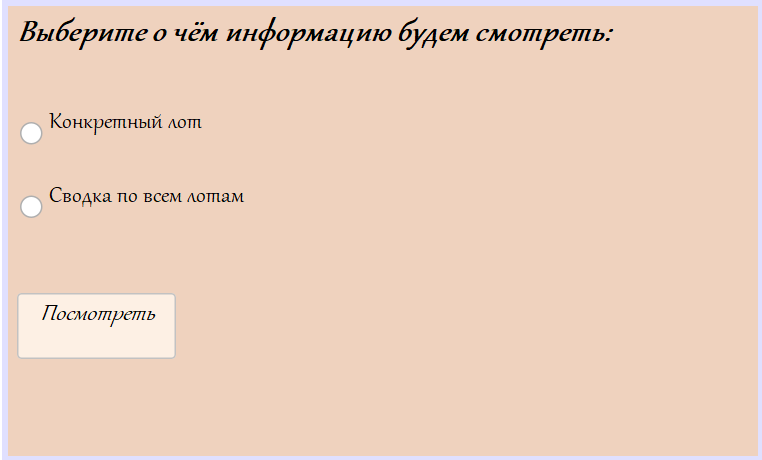
chooseExitOrNextDialog



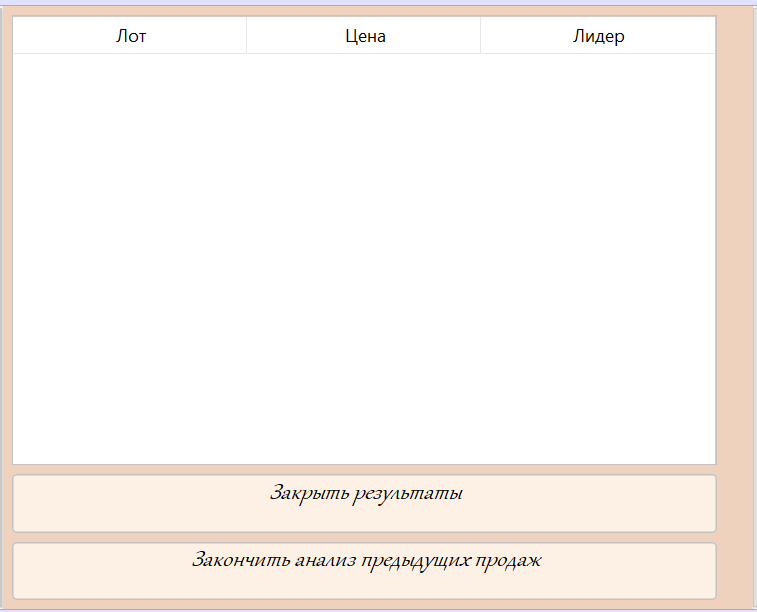
choosePrevLotDialog



oneTourDialog



chooseWhatLotSeeDialog



allLotsTableDialog

**7. Описание функционирования реализованной программы:**

В программе реализована имитация аукциона. Пользователь выступает в качестве аукциониста (человека, проводящего аукцион).

*Описание взаимодействия пользователя и программы:*

Пользователь может начать аукцион и выбрать лот для продажи. После нажатия кнопки «Далее» будет предложен выбор метода проведения продажи (автоматический или ручной). В автоматическом режиме выполняется имитация всех шагов аукциона, до продажи лота, аукционист может увидеть только окончательные параметры участников и финальную цену (динамику к ней). В ручном режиме аукционист может отслеживать параметры участников каждый ход (поднятие цены) и также увидеть окончательные параметры.

После продажи лота пользователь видит имя лидера и итоговую цену, также может выбрать такое действие, как посмотреть параметры участников – тогда он выбирает участника и далее выводится таблица с параметрами выбранного пользователя. После можно вернуться к завершающей странице и посмотреть параметры, например по другому участнику. Также есть возможность посмотреть динамику цены – от раунда к раунду. Если были проданные лоты до этого, то аукционисту становится видна кнопка «Посмотреть результаты предыдущих продаж». При просмотре предыдущих продаж можно выбрать просмотр статистики по конкретному лоту – тогда выведется имя лидера в том лоте, его цену и динамику цены, а можно выбрать сводку по всем лотам – в этом случае пользователь увидит таблицу с результатами всех прошедших до этого продаж (название лота, имя лидера и цену).

В конце пользователь может нажать «Завершить анализ продажи», в этом случае откроется окно с предложением продолжить аукцион – продать следующий лот, или завершить аукцион – выйти из приложения. В случае, если пользователь решает продолжить аукцион кнопка лота, который был продан исчезает. Если пользователь не выбрал ни один лот и хочет нажать далее – то никакого действия не будет.

*Описание внутреннего устройства программы:*

В программе при расчёте некоторых параметров использована библиотека jFuzzyLogic. Создано разбиение величин-параметров и написаны правила, в зависимости от которых выдается итоговый результат, расчёт искомого параметра.

В начале считывается информация из excel-файла о параметрах участников и лотов. Эти данные используются для старта аукциона, заполнения списков в классе MainDataOperations.

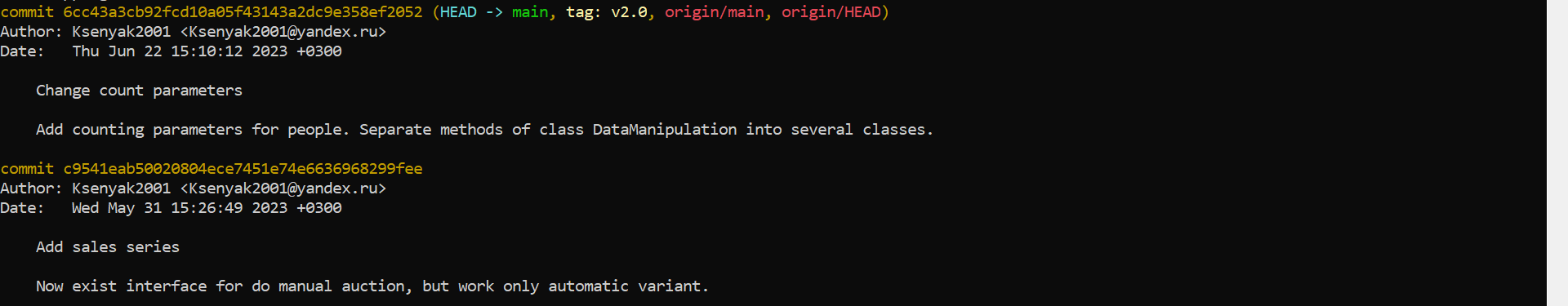
После выбора пользователем лота (с помощью JRadioButton) сохраняем информацию о выбранном номере. Скрываем кнопку, отвечающую за выбранный лот, чтобы исключить его выбор в дальнейшем.

Далее следует выбор режима проведения аукциона. В случае ручного режима программа выполняет один тур – одно поднятие цены, которое включает в себя расчёт предполагаемой цены, её приемлемости, эффектов аукциона, эмоций участников и величины, на которую поднимается цена. После идёт проверка на то, что не было определённого количества пустых ходов и в зависимости от результата проверки выводится окно с информацией о ходе аукциона. В автоматическом режиме программа прогоняет необходимое количество туров, до тех пор, пока не будет получено определённое количество пустых ходов.

По завершении продажи пересчитываются параметры участников (в ходе продажи сохраняются все необходимые для этого величины, например поднятие цены, количество ставок, максимальная ставка и другие), добавляется информация в объект класса HistoryManipulation (какой лот продан, за какую цену и кому). Далее если выбрано продолжение аукциона – продажа следующего лота, то начинается новый аукцион по схожему алгоритму. Добавляется отображение результатов предыдущих продаж по завершении текущей продажи.

Так возможно провести серию торгов до тех пор, пока аукционист не решит завершить аукцион.

*Приложения А. Логи системы контроля версий:*



*Приложение Б. Исходный код программы:*

<https://github.com/Ksenyak2001/EnglishAuction>